Active Learning of Mealy Machines with Timers

Véronique Bruyère, Bharat Garhewal, Guillermo A. Pérez,
Gaëtan Staquet, Frits W. Vaandrager

Theoretical computer science
University of Mons

Formal Techniques in Software Engineering
University of Antwerp

September 20, 2024

Many real-world scenarios have **timing** constraints, *e.g.*, giving a Highlights talk, network protocols.

[1] Angluin, "Learning Regular Sets from Queries and Counterexamples", 1987.

[2] Waga, "Active Learning of Deterministic Timed Automata with Myhill-Nerode Style Characterization", 2023.

Many real-world scenarios have **timing** constraints, *e.g.*, giving a Highlights talk, network protocols.

Well-known model for such systems: **timed Mealy machines** which are Mealy machines (*i.e.*, defining a function from input words to output words) augmented with **clocks**.

[1] Angluin, "Learning Regular Sets from Queries and Counterexamples", 1987.

[2] Waga, "Active Learning of Deterministic Timed Automata with Myhill-Nerode Style Characterization", 2023.

Many real-world scenarios have **timing** constraints, *e.g.*, giving a Highlights talk, network protocols.

Well-known model for such systems: **timed Mealy machines** which are Mealy machines (*i.e.*, defining a function from input words to output words) augmented with **clocks**.

Learning (in Angluin's[1] style) timed Mealy machines[2] is hard.
We focus on **timers** $\rightsquigarrow$ **Mealy machines with timers** (**MMT**).

[1] Angluin, "Learning Regular Sets from Queries and Counterexamples", 1987.

[2] Waga, "Active Learning of Deterministic Timed Automata with Myhill-Nerode Style Characterization", 2023.

Many real-world scenarios have **timing** constraints, *e.g.*, giving a Highlights talk, network protocols.

Well-known model for such systems: **timed Mealy machines** which are Mealy machines (*i.e.*, defining a function from input words to output words) augmented with **clocks**.

Learning (in Angluin's[1] style) timed Mealy machines[2] is hard.
We focus on **timers** $\rightsquigarrow$ **Mealy machines with timers** (**MMT**).

Clocks start at zero and grow towards infinity.

Timers start a given value and decrease towards zero.

[1] Angluin, "Learning Regular Sets from Queries and Counterexamples", 1987.

[2] Waga, "Active Learning of Deterministic Timed Automata with Myhill-Nerode Style Characterization", 2023.

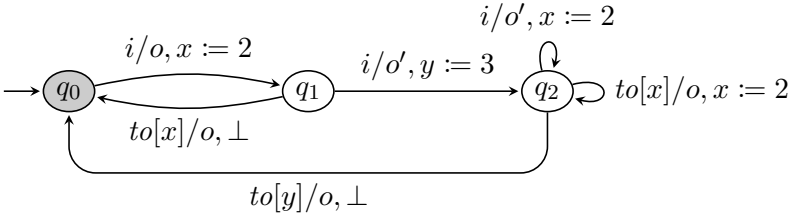

Figure 1: A Mealy machine with timers (**MMT**).

$(q_0, \emptyset)$

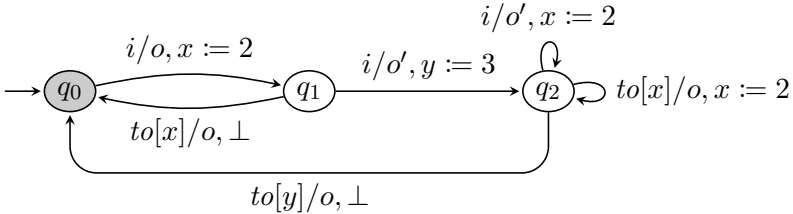

Figure 1: A Mealy machine with timers (**MMT**).

$$(q_0, \emptyset) \xrightarrow{0.5} (q_0, \emptyset)$$

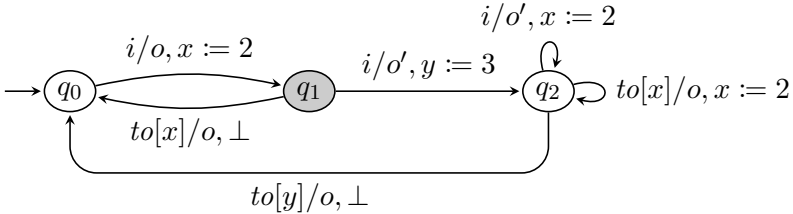

Figure 1: A Mealy machine with timers (**MMT**).

$$(q_0, \emptyset) \xrightarrow{0.5} (q_0, \emptyset) \xrightarrow{i/o} (q_1, x = 2)$$

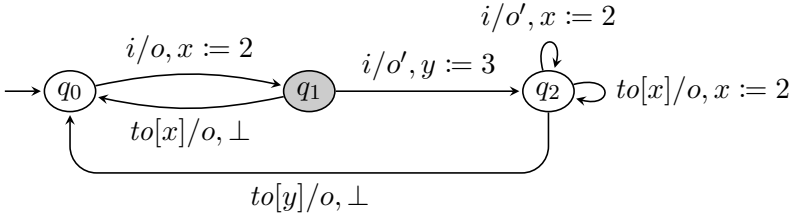

Figure 1: A Mealy machine with timers (**MMT**).

$$(q_0, \emptyset) \xrightarrow{0.5} (q_0, \emptyset) \xrightarrow{i/o} (q_1, x = 2) \xrightarrow{2} (q_1, x = 0)$$

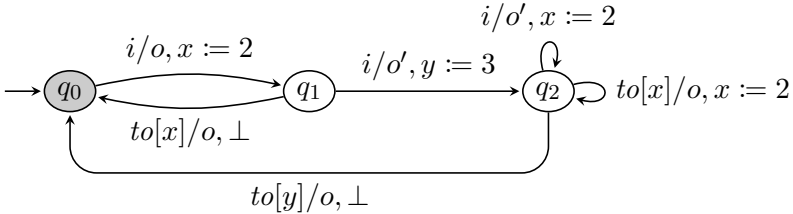

Figure 1: A Mealy machine with timers (**MMT**).

$$(q_0, \emptyset) \xrightarrow{0.5} (q_0, \emptyset) \xrightarrow{i/o} (q_1, x = 2) \xrightarrow{2} (q_1, x = 0) \xrightarrow{to[x]/o} (q_0, \emptyset)$$

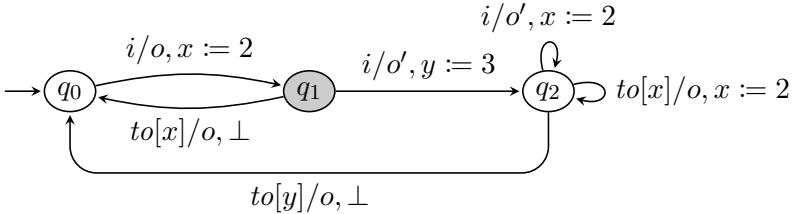

Figure 1: A Mealy machine with timers (**MMT**).

$$(q_0, \emptyset) \xrightarrow{0.5} (q_0, \emptyset) \xrightarrow{i/o} (q_1, x = 2) \xrightarrow{2} (q_1, x = 0) \xrightarrow{to[x]/o} (q_0, \emptyset)$$
$$\xrightarrow{0} (q_0, \emptyset) \xrightarrow{i/o} (q_1, x = 2)$$

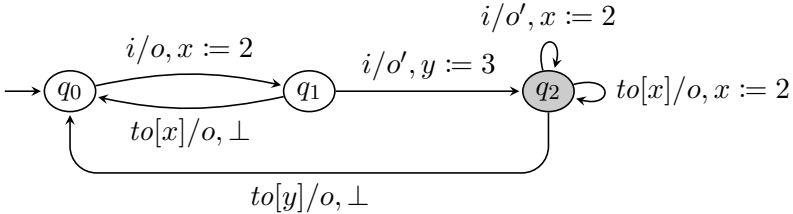

Figure 1: A Mealy machine with timers (**MMT**).

$$(q_0, \emptyset) \xrightarrow{0.5} (q_0, \emptyset) \xrightarrow{i/o} (q_1, x = 2) \xrightarrow{2} (q_1, x = 0) \xrightarrow{to[x]/o} (q_0, \emptyset)$$
$$\xrightarrow{0} (q_0, \emptyset) \xrightarrow{i/o} (q_1, x = 2) \xrightarrow{2} (q_1, x = 0) \xrightarrow{i/o'} (q_2, x = 0, y = 3)$$

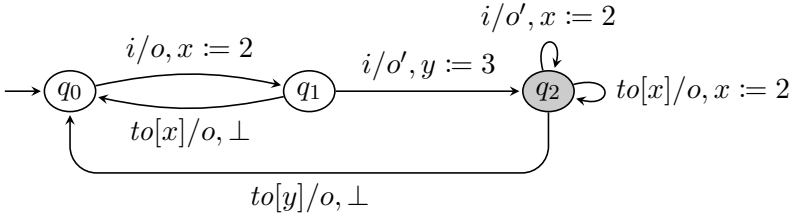

Figure 1: A Mealy machine with timers (**MMT**).

$$(q_0, \emptyset) \xrightarrow{0.5} (q_0, \emptyset) \xrightarrow{i/o} (q_1, x = 2) \xrightarrow{2} (q_1, x = 0) \xrightarrow{to[x]/o} (q_0, \emptyset)$$
$$\xrightarrow{0} (q_0, \emptyset) \xrightarrow{i/o} (q_1, x = 2) \xrightarrow{2} (q_1, x = 0) \xrightarrow{i/o'} (q_2, x = 0, y = 3)$$
$$\xrightarrow{0} (q_2, x = 0, y = 3).$$

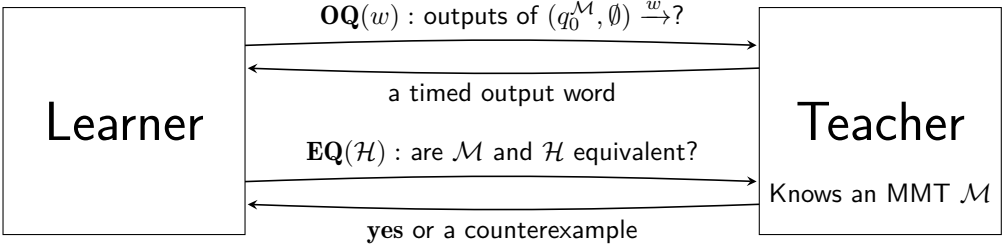

Figure 2: Adaptation of Angluin's framework[3] to MMTs.

[3] Angluin, "Learning Regular Sets from Queries and Counterexamples", 1987.

We adapt $L^\#$ (active learning algorithm for Mealy machines[4]) to MMTs: $L^\#_{\text{MMT}}$.

> ***Theorem 1.*** *Let $\mathcal{M}$ be a* ***good*** *MMT and ℓ be the length of the longest counterexample. Our $L^\#_{MMT}$ learning algorithm requires a number of queries* ***polynomial*** *in the number of states of $\mathcal{M}$ and in ℓ, and* ***exponential*** *in the number of timers of $\mathcal{M}$.*

[4] Vaandrager et al., "A New Approach for Active Automata Learning Based on Apartness", 2022.
[5] Bruyère, Pérez, et al., "Automata with Timers", 2023.

We adapt $L^\#$ (active learning algorithm for Mealy machines[4]) to MMTs: $L^\#_{\text{MMT}}$.

Theorem 1. *Let $\mathcal{M}$ be a* ***good*** *MMT and ℓ be the length of the longest counterexample. Our $L^\#_{MMT}$ learning algorithm requires a number of queries* ***polynomial*** *in the number of states of $\mathcal{M}$ and in ℓ, and* ***exponential*** *in the number of timers of $\mathcal{M}$.*

Problem. There may be actions that occur **simultaneously**, due to **timeouts**.

$\hookrightarrow$ A **good** MMT ensures that we can observe every behavior by changing some delays.[5]

[4] Vaandrager et al., "A New Approach for Active Automata Learning Based on Apartness", 2022.
[5] Bruyère, Pérez, et al., "Automata with Timers", 2023.

Thank you!

For all details, see
Bruyère, Pérez, et al., "Automata with Timers", 2023, and
Bruyère, Garhewal, et al., "Active Learning of Mealy Machines with Timers", 2024.
For our implementation, see
`https://doi.org/10.5281/zenodo.10647628`, and
`https://gitlab.science.ru.nl/bharat/mmt_lsharp`.

References I

- Angluin, Dana. "Learning Regular Sets from Queries and Counterexamples". In: *Inf. Comput.* 75.2 (1987), pp. 87–106. DOI: 10.1016/0890-5401(87)90052-6.
- Bruyère, Véronique, Bharat Garhewal, et al. "Active Learning of Mealy Machines with Timers". In: *CoRR* abs/2403.02019 (2024). DOI: 10.48550/arXiv.2403.02019. arXiv: 2403.02019.
- Bruyère, Véronique, Guillermo A. Pérez, et al. "Automata with Timers". In: *Formal Modeling and Analysis of Timed Systems - 21st International Conference, FORMATS 2023, Antwerp, Belgium, September 19-21, 2023, Proceedings*. Ed. by Laure Petrucci and Jeremy Sproston. Vol. 14138. Lecture Notes in Computer Science. Springer, 2023, pp. 33–49. DOI: 10.1007/978-3-031-42626-1_3.

References II

- Vaandrager, Frits W. et al. "A New Approach for Active Automata Learning Based on Apartness". In: *Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2022, Munich, Germany, April 2-7, 2022, Proceedings, Part I*. Ed. by Dana Fisman and Grigore Rosu. Vol. 13243. Lecture Notes in Computer Science. Springer, 2022, pp. 223–243. DOI: 10.1007/978-3-030-99524-9_12.
- Waga, Masaki. "Active Learning of Deterministic Timed Automata with Myhill-Nerode Style Characterization". In: *Computer Aided Verification - 35th International Conference, CAV 2023, Paris, France, July 17-22, 2023, Proceedings, Part I*. Ed. by Constantin Enea and Akash Lal. Vol. 13964. Lecture Notes in Computer Science. Springer, 2023, pp. 3–26. DOI: 10.1007/978-3-031-37706-8_1.